

基于协同进化的混合变量多目标粒子群优化算法求解无人机协同多任务分配问题

王峰^{1)*} 张衡¹⁾ 韩孟臣¹⁾ 邢立宁²⁾

¹⁾(武汉大学 计算机学院, 武汉 430072)

²⁾(国防科技大学 系统工程学院, 长沙 410073)

摘要 无人机多机协同控制系统近年来已被广泛地应用在军事打击、海洋监测、陆地航拍和灾情探测等领域。针对无人机协同多任务分配问题, 为了更加准确地描述无人机协同多任务分配场景, 本文考虑实际应用场景下的多种复杂约束, 并以无人机飞行总航程最少和任务完成时间最短为优化目标, 构建了混合变量多约束的无人机协同多任务分配问题模型 M-CMTAP。为了高效求解上述模型, 本文提出一种基于协同进化的混合变量多目标粒子群优化算法 C-MOPSO。C-MOPSO 采用基于任务分配和路径规划的编码方法表示无人机的任务分配结果和路径规划结果及基于约束处理的可行解初始化方法生成可行粒子; 同时利用基于结构学习的重组策略对粒子进行更新以提高种群的多样性和收敛性; 并引入协同进化策略在两个子种群之间进行合作进化以提高算法的搜索效率。根据无人机和目标分布状态设计 4 个代表性的测试实例并验证算法性能, 实验结果表明, 与其他采用协同进化策略的算法相比, 所提算法在解的收敛性和解集多样性上均具有显著的性能优势。
关键词 协同进化; 粒子群优化算法; 混合变量优化问题; 多目标优化; 无人机任务分配问题
中图法分类号 TP18

Co-evolution based Mixed-Variable Multi-Objective Particle Swarm Optimization for UAV Cooperative Multi-Task Allocation Problem

Wang Feng^{1)*} Zhang Heng¹⁾ Han Meng-Chen¹⁾ Xing Li-Ning²⁾

¹⁾(School of Computer Science, Wuhan University, Wuhan, 430072)

²⁾(College of System Engineering, National University of Defense Technology, Changsha, 410073)

Abstract In recent years, the multi-aircraft cooperative control system of UAV has been widely used in many application fields, such as military strike, ocean monitoring, aerial photography on land and disaster detection. In order to describe UAV cooperative multi-task allocation scenarios more accurately, this paper proposes a UAV cooperative multi-task allocation model (M-CMTAP) that considers multiple constraints and multiple optimization objectives simultaneously, which involves mixed decision variables and multiple constraints. These constraints describe the relationships in the complex decision variables in the actual military scenarios, including UAV resource constraint, UAV type constraint, mission execution sequence constraint and multi-aircraft cooperation constraint, etc. In M-CMTAP, two optimization objectives are taken into account, i.e., the total flight range of the UAVs and the completion time for all missions. The shortest flight range of the UAVs means that the UAVs consume the least flight resources during the mission, while the minimum completion time for all missions ensures that the entire military mission can be completed quickly. In order to solve this M-CMTAP model more efficiently, this paper proposes a co-evolution based mixed-variable multi-objective particle swarm optimization algorithm named C-MOPSO. C-MOPSO firstly adopts the mixed variable coding method to

本课题得到国家自然科学基金(61773296, 61773120)、高等学校全国优秀博士学位论文作者专项资金(2014-92)、广东省普通高校创新团队项目(2018KCXTD031)资助。王峰(通信作者), 博士, 副教授, 中国计算机学会(CCF)高级会员(会员号: 12417S), 主要研究领域为进化计算、机器学习与智能信息处理. E-mail: fengwang@whu.edu.cn. 张衡, 硕士研究生, 主要研究领域为进化计算. E-mail: zhangheng96@whu.edu.cn. 韩孟臣, 硕士研究生, 主要研究领域为进化计算. E-mail: hanmengchen@whu.edu.cn. 邢立宁, 博士, 研究员, 主要研究领域为优化理论与应用. E-mail: xln_2002@nudt.edu.cn.

represent the task allocation, where some discrete variables represent the allocation relationship between the tasks and the UAVs, while some continuous variables and some other discrete variables represent the resource consumption of the UAVs during the mission. In order to generate feasible particles satisfying various constraints, a feasible solution initialization method with constraint processing is further proposed. And a structure learning based reproduction method is then employed to update to update the particles to improve the diversity and convergence of the population. The structure learning based reproduction method learns the historical structural information of good solutions and the corresponding structure length can help keep good convergence and diversity of the population. After getting the position vector of the good solutions, it sequentially adds new task numbers, UAV numbers and resource consumption values satisfying the constraints, until all tasks have been allocated to the UAVs and completed. At the same time, in order to further improve the search efficiency of the algorithm, this paper introduces the idea of co-evolution to design the search strategy, and different populations exchange search information through the way of co-evolution. In C-MOPSO, the co-population co-evolves with the particle swarm, and the fast non-dominant sorting method is also adopted to select the first N dominant particles from the fusion population which composed by the co-population and the particle swarm to update the Pbest. Due to the co-evolution based population search strategy, it can balance the diversity and convergence of the population well, which can help to generate better particles, so as to obtain better solutions. In order to verify the effectiveness of C-MOPSO, four representative M-CMTAP test cases are designed based on different UAV distributions and mission distributions. The experimental results on the four representative test problems show that, compared with some state-of-the-art co-evolution based algorithms, the proposed algorithm C-MOPSO outperforms others on convergence efficiency and diversity of solution sets.

Key words co-evolution; particle swarm optimization; mixed-variable optimization problem; multi-objective optimization; UAV cooperative multi-task assignment problem

1 引言

无人机 (Unmanned Aerial Vehicle, UAV) 是指没有机上人员操控、通过路基或海基无线电遥控设备远程控制的飞行器^[1]。相比有人飞行器, 无人机主要有以下两个方面的优势。一方面, 无人机飞行无需考虑飞行员的生理极限, 可以执行超高空、长距离和危险地区侦察与打击等任务; 另一方面, 无人机无需提供装备人类生命支持系统, 有效载荷明显增多。近年来, 得益于无人机技术的发展, 无人机已在军事和民用科研领域被广泛应用, 例如军事打击、海洋监测、陆地航拍和灾情探测等领域。

随着军事领域的不断发展, 由多架无人机协同工作的无人机系统 (Unmanned Aerial System, UAS) 已被各国用来执行军事任务。在 UAS 中, 多架无人机通过协同合作分别对多个军事目标进行观测、打击和评估任务, 最终完成整个军事行动。虽然多架无人机协同作战可以高效地完成的任务, 但是控制多架无人机也给指挥基站带来了新的挑战。一方面, 操控人员手动控制多无人机会占用大量人力资源,

因此需要采用自动控制系统替代人力; 另一方面, 无人机协同作战时的任务分配策略决定了任务完成的质量, 一个合理的任务分配计划能够帮助无人机更加高效地完成的任务。因此, 为了高效地制定无人机的任务分配计划、科学地控制无人机, 研究者们基于无人机系统的工作场景提出了无人机协同多任务分配模型 (Cooperative Multiple Task Allocation Problem, CMTAP)^[2]。

CMTAP 模型可有效刻画将多个任务分配给无人机系统以及无人机协同执行任务的场景。目前的 CMTAP 模型通常都考虑了多种约束条件。例如, Huang 等人提出了考虑任务资源需求约束的同构无人机协同任务分配模型^[3], Cheng 等人提出了考虑传感器约束、时间窗约束、燃料成本约束和危险程度约束的同构无人机任务分配模型^[4], Ye 等人采用一种自适应的遗传算法求解考虑任务优先级约束的异构无人机模型^[5], Guang 等人提出了考虑无人机运动学约束的异构无人机模型^[6]。然而, 随着实际场景变得更加复杂, 单个无人机通常无法独立完成一项任务, 需要多架无人机协同工作以保证任务顺利完成, 因此, 模型还需要考虑多架无人机协同

工作时的约束条件，如多机协同约束、任务完成需求约束和任务执行时序约束等约束条件。

在实际应用中，建立无人机协同多任务分配模型还应根据不同的行动需求来构建不同的优化目标。根据优化目标数目的不同，可将现有的无人机协同多任务分配模型分为单目标模型和多目标模型。苏菲等人提出的单目标模型以任务分配数量作为任务分配的目标^[7]，梁国强等人提出的单目标模型将无人机飞行总航程作为优化目标^[8]，王福星等人则采用了无人机最长暴露时间指标评估任务分配结果的优劣^[9]。Cao 等人提出的模型假设无人机执行任务的收益相同，以最小化所有无人机的逻辑飞行距离为优化目标^[10]。邸斌等人提出了包含异构无人机的分布式 CMTAP 模型，并采用无人机执行任务收益作为模型的优化目标^[11]。王健等人提出了带有专家信度的单目标模型，该模型将任务分配过程中的风险作为模型的优化目标^[12]。另外，一些模型同时考虑了多个优化目标，例如，Guang 等人提出的模型同时考虑了最大执行时间和无人机累积执行时间两个优化目标^[6]，韩博文等人提出了同时考虑无人机任务收益、任务执行时间和任务执行代价的多目标模型^[13]，田震等人采用遗传算法求解了考虑任务执行时间和总体加权攻击收益的异构无人机任务分配模型^[4]。由于无人机系统的任务需求复杂且多变，仅仅用一个指标评估任务分配计划的质量往往无法给决策者提供更有效的决策信息。因此，模型通常需要采用多种指标对任务分配计划的质量进行综合评估。

基于上述分析，本文建立了一个多约束多目标的无人机协同多任务分配模型 M-CMTAP。该模型考虑了实际军事场景下的多种实际约束，如无人机资源约束、无人机类型约束、任务执行时序约束和多机协同约束等约束条件。此外，为了更加准确地评估任务分配计划的质量，该模型同时考虑了无人机飞行总航程和任务完成时间两个目标。其中，无人机飞行总航程用于衡量无人机完成任务时的资源消耗程度，任务完成时间则用于评估整个无人机系统的工作效率。

M-CMTAP 模型不仅包含混合变量，同时还存在复杂的多个约束条件，因此，传统的多目标优化算法并不能有效地对问题空间进行搜索并生成满足多种约束条件的可行解。混合变量和多种约束使得问题空间变得不规则，增加了问题空间的复杂度，提高了模型求解难度。为了有效求解上述模型，

本文提出了一种基于协同进化的混合变量多目标粒子群优化算法 C-MOPSO。C-MOPSO 采用了混合变量编码方法表示任务分配情况，其中，离散变量表示任务和无人机之间的分配关系，连续变量和离散变量表示无人机在执行任务时的资源消耗情况。为了生成满足多种约束的可行粒子，本文进一步提出了基于约束处理的可行解初始化方法。同时，为了提高算法的搜索效率，本文引入协同进化的思想设计搜索策略^[15]，不同种群之间将通过协同进化的方式交换搜索信息。为了验证 C-MOPSO 的有效性，本文基于不同的无人机分布状态和任务分布状态，设计了 4 个 M-CMTAP 测试实例，并基于该测试实例对 C-MOPSO 算法的有效性进行了验证。

2 无人机协同多任务分配模型

通常一个无人机协同多任务分配模型可表示为一个四元组 $\langle U, T, M, C \rangle$ ，其中， U 为无人机集合， T 为军事区域内存在的目标集合， M 是所有目标上的任务集合， C 为问题模型中的约束集合。无人机、目标和任务是 CMTAP 的三个重要对象。本文在传统 CMTAP 模型基础上，进一步考虑实际军事场景的多个约束条件和优化目标，构建了基于多约束多目标的无人机协同多任务分配模型 M-CMTAP，表 1 给出了该模型中的三个对象属性。

M-CMTAP 模型主要考虑以下三种不同类型的无人机：执行打击任务的战斗无人机 (Fighter) 和执行观测任务的观测无人机 (Observer) 和打击结果评估任务的侦察无人机 (Scout)。

设无人机集合 U 的前 N_s 架为侦察无人机，后 N_f 架为战斗无人机，执行任务前，所有无人机都在军事区域内的一个固定位置等待。每个目标都包括了三种类型的任务：观测任务 (Observe)、打击任务 (Attack) 和打击结果评估任务 (Evaluate)，则有： $N_m = 3N_t$ 。对于不同类型的任务，无人机完成任务都需要消耗不同数量和类型的资源。例如，完成目标 T_j 的打击任务需要消耗一定数量的战斗无人机机载导弹数目；完成目标的观测或评估则需要侦察无人机在目标位置处侦察一定时间。当执行该任务的所有无人机消耗总资源满足完成任务所需资源 (AR_k, OR_k, ER_k) 时，任务顺利完成。当所有目标的任务全部完成时，整个军事任务完成。

表 1 M-CMTAP 中的对象属性及符号解释

	属性
无人机 U	侦察无人机总数 N_S
	战斗无人机总数 N_F
	无人机总架数 N_U
	初始位置 $P_i = (x_i, y_i)$
	飞行速度 V_i
	无人机类别 $UT_i \in \{Fighter, Scout\}$
	战斗无人机最大携弹数目 $Load_i$
	无人机最远飞行距离 $MaxDis_i$
目标 T	目标个数 N_T
	位置坐标 $P_i = (x_i, y_i)$
	目标任务个数 $N_T^j = 3$
任务 M	任务总个数 N_m
	类型 $MT_k \in \{Observe, Attack, Evaluate\}$
	观测类型任务完成所需观测时间 OR_k
	打击类型任务完成所需导弹数目 AR_k
	评估类型任务完成所需评估时间 ER_k

在保证模型有效的前提下，为简化模型并降低模型求解难度，本文对模型提出了如下合理假设：

- 1) 在任务分配时，所有目标的地理坐标对于指挥中心是已知的。
- 2) 无人机在执行任务时保持匀速直线飞行。
- 3) 同一观测任务或评估任务可由多架侦察无人机协同完成，当无人机的总执行时间满足任务完成所需观测或评估时间时，任务即完成。
- 4) 同一打击任务可由多架战斗无人机协同完成，战斗无人机的打击总能成功。当多架战斗无人机执行打击任务时消耗的导弹数目大于任务完成所需的导弹数目时，打击任务完成。

5) 针对某个目标的军事任务全部完成当且仅针对该目标的打击结果评估任务完成。

6) 无人机 U_i 执行任务 M_k 时，如果任务 M_k 的前序任务未完成，则无人机在任务上空保持等待并以速度 V_i 环绕飞行。

7) 执行观测任务或评估任务时，侦察无人机在目标上空以速度 V_i 飞行，执行打击任务的战斗无人机在完成打击任务后即可离开任务执行地点。

基于以上符号解释和模型假设，可将用无人机协同的多任务分配计划表示为一个向量，如表 2 所示。无人机任务分配计划向量共有 4 行。第 1 行表示目标序列，第 2 行表示目标上的任务序列，第 3 行和第 4 行表示问题的决策向量，其中第 3 行表示任务分配结果，第 4 行表示无人机执行分配任务所需的资源消耗量。

需要指出的是，由于不同的无人机可执行的任务类型是固定的，因此可以通过分析无人机任务分配计划向量中不同变量之间的关系来确定相关参数的取值范围。如第 2 行中，任务序列的下标第一部分的取值表示不同任务类型， $M_{K,O}$ 、 $M_{K+1,A}$ 、 $M_{K+2,E}$ 分别为目标 T_j 的观测任务、打击任务和打击结果评估任务。第 3 行中， X_i^j 取值为 1 则表示任务 j 被分配给了第 i 架无人机执行，当 X_i^K 和 X_i^{K+2} 取值为 1 时，表示观测任务 $M_{K,O}$ 和打击结果评估任务 $M_{K+2,E}$ 都被分配给第 i 架无人机执行。而在实际军事作战过程中，观测任务和打击结果评估任务只能由侦察无人机执行，因此可以确定 i 的取值范围为 $[1, N_S]$ 。 X_i^{K+1} 取值 1 时表示打击任务被分配给了第 i 架无人机执行，由于打击任务只能选择战斗无人机执行，因此 i 的取值范围为 $[N_S+1, N_U]$ 。第 4 行中，当任务类型为观测任务或打击结果评估任务时， C_i^K 为连续变量，表示无人机 U_i 执行任务 M_k 消耗的时间资源数量；当任务类型为打击任务时， C_i^K 为离散变量，表示无人机执行该打击任务时需要消耗的导弹资源数量。

表 2 无人机任务分配计划向量

T_j														
$M_{K,O}$					$M_{K+1,A}$					$M_{K+2,E}$				
X_1^K	...	X_i^K	...	$X_{N_S}^K$	$X_{N_S+1}^{K+1}$...	X_i^{K+1}	...	$X_{N_U}^{K+1}$	X_1^{K+2}	...	X_i^{K+2}	...	$X_{N_S}^{K+2}$
C_1^K	...	C_i^K		$C_{N_S}^K$	$C_{N_S+1}^{K+1}$...	C_i^{K+1}	...	$C_{N_U}^{K+1}$	C_1^{K+2}	...	C_i^{K+2}		$C_{N_S}^{K+2}$

表 3 展示了一个包含 2 个目标，3 架侦察无人机和 4 架战斗无人机的 M-CMTAP 对应的一个任务分配计划实例。从表 3 中可以看到，第一个目标的

观测任务被分配给第 1 架和第 2 架侦察无人机执行，两架无人机执行目标 T_1 的观测任务时一共消耗了 30 个单位时间。对第一个目标的打击任务则由

第1架和第3架战斗无人机协同完成，两架无人机在执行该任务时一共消耗了4枚导弹。

由于决策变量之间存在多个复杂约束，在实际军事作战过程中，生成任务分配计划需要充分考虑

模型变量之间的约束关系。为了保证任务分配计划科学合理，在确定无人机的任务分配和资源消耗时需要考虑多种实际的约束条件。

表3 无人机任务分配计划实例

T_1									T_2										
$M_{1,O}$			$M_{2,A}$			$M_{3,E}$			$M_{4,O}$			$M_{5,A}$			$M_{6,E}$				
1	0	1	1	0	1	0	1	1	0	0	1	1	1	0	1	1	1	1	1
25.48	0	5.52	1	0	3	0	3.02	6.98	0	0	34.57	23.43	3	0	2	3	1.23	4.64	7.13

2.1 模型的约束条件

M-CMTAP模型考虑了6种不同的约束条件。

1) 战斗无人机最大携弹数目约束。受限于无人机的运载能力和无人机的结构设计，战斗无人机执行打击任务时最多只能挂载一定数目的导弹。因此，单个战斗无人机执行打击任务消耗的导弹数要小于其最大携弹数目。

$$\sum_{k=1}^{Nm} C_i^k X_i^k \leq Load_i \quad (1)$$

其中 $1 \leq k \leq Nm$ 且 $MT_k \in \{Attack\}$ 。

2) 无人机类型约束。侦察无人机只能完成观测任务和评估任务，而战斗无人机只能完成打击任务。在为无人机分配任务时，需要满足无人机类型约束。

$$X_i^k = \begin{cases} 1 \text{ or } 0 & (p1 \wedge q1) \vee (p2 \wedge q2) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where

$$p1: MT_k \in \{Observe, Evaluate\}$$

$$q1: UT_i \in \{Scout\}$$

$$p2: MT_k \in \{Attack\}$$

$$q2: UT_i \in \{Fighter\}$$

其中 $1 \leq i \leq N_S + N_F$, $1 \leq k \leq Nm$ 。很明显，只有当任务类型与无人机的类型匹配时，任务分配结果变量 X_i^k 才有可能取值为1。

3) 任务执行时序约束。军事任务要求打击任务在观测任务完成后才能开始执行，而打击结果评估任务需要在打击任务完成后开始。对于第 j 个目标上的任务，任务执行时序约束表示如下：

$$t_k^e \leq t_{k+1}^s \leq t_{k+1}^e \leq t_{k+2}^s \quad (3)$$

其中， $k = 3(j-1) + 1$ 且 $1 \leq j \leq Nt$ 。 t_k^e 为观测任务的结束时间， t_{k+1}^s 和 t_{k+1}^e 分别为打击任务的开始和结束时间， t_{k+2}^s 则为评估任务的开始时间。

4) 多机协同约束。为了保证针对目标的军事任务顺利完成，该目标的每种类型的任务需要分配给至少一架无人机执行，即有：

$$\sum_{i=1}^S X_i^k \wedge \sum_{i=S+1}^{S+F} X_i^{k+1} \wedge \sum_{i=1}^S X_i^{k+2} = 1 \quad (4)$$

5) 任务需求约束。为了保证每个任务被成功完成，执行任务的无人机消耗的资源需要满足任务完成所要求的条件。

对于打击任务，执行该打击任务的战斗无人机消耗的导弹总数需要大于该任务完成所需导弹数：

$$\sum_{i=S+1}^{S+F} C_i^k X_i^k \geq AR_k \quad UT_i = Fighter, MT_k = Attack \quad (5)$$

对于观测任务和打击结果评估任务，侦察无人机执行任务的时间消耗需要大于该任务完成所需执行时间：

$$\sum_{i=1}^S C_i^k X_i^k \geq OR_k \quad UT_i = Scout, MT_k = Observe \quad (6)$$

$$\sum_{i=1}^S C_i^k X_i^k \geq ER_k \quad UT_i = Scout, MT_k = Evaluate \quad (7)$$

6) 最大航程约束。无人机执行任务时的总飞行距离应小于无人机的最远飞行距离。无人机 U_i 的任务执行序列为 $Seq^i = \{M_1^i, M_2^i, \dots, M_{N_i}^i\}$ ， $Dis_{M_k^i}^{M_{k+1}^i}$ 表示无人机 U_i 执行的第 k 个任务到第 $k+1$ 个任务的飞行距离。即：

$$\sum_{M_k^i \in Seq^i} Dis_{M_k^i}^{M_{k+1}^i} \leq MaxDis_i \quad (8)$$

2.2 模型的目标函数

为了使无人机任务分配计划更加科学合理，本文构建的M-CMTAP模型同时考虑无人机飞行总航程和任务完成时间两个优化指标。

无人机飞行总航程描述了所有无人机在完成时的飞行路径长度，距离越短，表示该任务分配集合所需飞行资源消耗越少；反之则表示该任务分配集合需消耗较多飞行资源。无人机执行任务时的飞行总航程最短可以保证无人机执行任务时消耗的飞行资源最少。设无人机 U_i 依次执行的任务序列为

$Seq^i = \{M_1^i, M_2^i, \dots, M_{N_i}^i\}$ ， $Dis_{M_k^i}^{M_{k+1}^i}$ 为无人机 U_i 执行的第

k 个任务到第 $k+1$ 个任务的飞行距离, 则无人机飞行总航程为:

$$f1 = \sum_{i=1}^{Nu} \sum_{M_k^i \in Seq^i} Dis_{M_k^i}^{M_{k+1}^i} \quad (9)$$

为了尽可能保证任务较快完成, 该模型还考虑以所有任务的完成时间为另一个优化目标, 所有任务的完成时间最短可以保证整个任务能被快速执行完毕。任务完成时间可表示如下:

$$f2 = \max_{1 \leq k \leq Nm} t_k^e \quad (10)$$

其中, t_k^e 表示任务 M_k 的完成时间。所有目标任务的完成时间即代表军事行动的任务完成时间。

由于分配任务时希望无人机尽快完成任务的同时消耗较少地飞行资源, 因此, 本文提出的模型同时考虑了上述的两个优化目标, 即同时最小化函数 $f1$ 和 $f2$ 。

无人机航程和其消耗资源数紧密相关, 无人机飞行总航程越短, 代表无人机执行任务时消耗的资源越少。然而, 飞行总航程最短并不一定能保证军事任务的完成时间最短。例如, 一些无人机可能能在同一目标上分配较多的任务以满足所有无人机飞行总航程最短, 而这会导致该无人机完成所有任务的时间变长, 从而使得整个军事任务完成时间变长。因此, 无人机飞行总航程和任务完成时间两个指标存在一定的冲突。

2.3 M-CMTAP模型

根据以上分析, 本文建立基于多约束条件的多目标优化模型 M-CMTAP:

$$\begin{cases} \min & f1 \\ \min & f2 \end{cases} \quad (11)$$

s.t.

$$\begin{aligned} & \sum_{k=1}^{Nm} C_i^k X_i^k \leq Load_i \\ & X_i^k = \begin{cases} 1 \text{ or } 0 & (p1 \wedge q1) \vee (p2 \wedge q2) \\ 0 & \text{otherwise} \end{cases} \\ & p1: MT_k \in \{Observe, Evaluate\} \\ & q1: UT_i \in \{Scout\} \\ & p2: MT_k \in \{Attack\} \\ & q2: UT_i \in \{Fighter\} \\ & t_k^e \leq t_{k+1}^s \leq t_{k+1}^e \leq t_{k+2}^s \\ & \sum_{i=1}^S X_i^k \wedge \sum_{i=S+1}^{S+F} X_i^{k+1} \wedge \sum_{i=1}^S X_i^{k+2} = 1 \\ & \sum_{i=S+1}^{S+F} C_i^k X_i^k \geq AR_k \quad UT_i = Fighter, MT_k = Attack \end{aligned}$$

$$\begin{aligned} & t_k^e \leq t_{k+1}^s \leq t_{k+1}^e \leq t_{k+2}^s \\ & \sum_{i=1}^S X_i^k \wedge \sum_{i=S+1}^{S+F} X_i^{k+1} \wedge \sum_{i=1}^S X_i^{k+2} = 1 \\ & \sum_{i=S+1}^{S+F} C_i^k X_i^k \geq AR_k \quad UT_i = Fighter, MT_k = Attack \\ & \sum_{i=1}^S C_i^k X_i^k \geq OR_k \quad UT_i = Scout, MT_k = Observe \\ & \sum_{i=1}^S C_i^k X_i^k \geq ER_k \quad UT_i = Scout, MT_k = Evaluate \\ & \sum_{M_k^i \in Seq^i} Dis_{M_k^i}^{M_{k+1}^i} \leq MaxDis_i \end{aligned}$$

上述模型所包含变量既有表示任务分配结果的离散变量, 也有表示无人机资源消耗的离散和连续变量, 这些混合变量增加了问题空间的复杂度, 提高了算法搜索的难度。此外, 该模型包含多种约束条件, 既有不等式约束如战斗无人机最大携弹数目约束、无人机最大航程约束; 也有等式约束如多机协同约束, 这些约束条件使得问题对应的解空间变得不规则, 增加了算法搜索到可行解的难度。

由于现有的算法无法对 M-CMTAP 模型进行有效求解, 因此, 本文基于协同进化策略^[10]提出了一种基于协同进化的多目标粒子群优化算法 (Coevolution based Multi-objective Optimization Particle Swarm Optimization, C-MOPSO)。该算法采用混合变量编码的策略处理混合变量, 并利用特殊的约束处理方法生成可行解。

3 基于协同进化的混合变量多目标

粒子群优化算法

在 C-MOPSO 中, 针对混合变量, 采用了一种基于任务分配和路径规划的混合编码方法, 粒子的位置向量包含表示任务分配结果的离散变量和表示资源消耗的混合变量。为了更加有效地生成可行的任务分配计划, C-MOPSO 根据无人机任务分配问题的特点, 采用了基于任务分配和路径规划的混合变量编码方法, 并根据该编码方法设计合适的约束处理方法对粒子的位置向量进行初始化, 同时使用基于结构学习的重组方法生成新粒子以进一步提高种群的质量。此外, 在种群更新过程中, 为了获得更加优秀的粒子, C-MOPSO 采用了协同进化机制, 即通过一个协同进化的种群与原有种群的竞争协同方式来生成新粒子。

3.1 基于任务分配和路径规划的混合变量编码方法

目前已有的表示任务分配计划的编码方法只能描述无人机执行单个任务时的信息^[8]，并不能表示无人机执行多个任务时的顺序信息，即无人机执行任务时的路径规划结果。因此，为了同时表示任务的分配结果和无人机执行任务时的路径规划结果，本文提出了一个基于任务分配和路径规划的混合变量编码方法。该方法根据无人机在执行任务时的具体情况进行编码，每个粒子的编码包含三个部分，即 $P=(MN,UN,C)$ 。其中，第一部分 MN 是离散变量，记录任务编号；第二部分 UN 是离散变量，记录执行对应任务的无人机编号；第三部分 C 是混合变量，记录无人机执行任务时的资源消耗。

图1展示了M-CMTAP模型对应的一个粒子的编码。对于一个包含2个目标，1架战斗无人机和1架侦察无人机的M-CMTAP模型，目标 T_1 上存在三种任务(M_1, M_2, M_3)，目标 T_2 上存在三种任务(M_4, M_5, M_6)，无人机 U_1 为侦察无人机，无人机 U_2 为战斗无人机。粒子的位置向量的前两行表示了任务分配结果。任务 M_1, M_3, M_4 和 M_6 被分配给无人机 U_1 执行。此时，对应的任务分配计划向量中的 $X_1^k=1$ ($k \in \{1,3,4,6\}$)。任务 M_2 和 M_5 被分配给了无人机 U_2 执行，此时，对应的任务分配计划向量中的 $X_1^k=1$ ($k \in \{2,5\}$)。粒子的位置向量的第三行则表示了无人机执行每个任务时的资源消耗结果，该向量中既有表示侦察无人机执行观测任务和打击结果评估任务时间消耗的连续变量，也有表示战斗无人机执行打击任务消耗导弹数目的离散变量。

任务编号MN	1	4	2	5	6	3
无人机任务编号UN	1	1	2	2	1	1
资源消耗C	25.5	13.5	3	2	16.5	5.5

图1 粒子编码

此外，为了表示无人机的路径规划结果，粒子的位置向量中任务编号出现的顺序代表了无人机执行任务的顺序。图1中黑色实心箭头表示无人机 U_1 执行任务依次为 M_1, M_4, M_6, M_3 ，黑色空心箭头表示无人机 U_2 的执行任务依次为 M_2, M_5 。

由于该编码方法采用基于任务分配和路径规划的混合变量编码方式，因此它不仅可以将粒子的位置向量高效地转化为M-CMTAP的一个任务分配计划，同时还可以充分考虑无人机任务执行过程中的路径不确定性，利用任务在粒子位置向量中的顺序显式地表示出无人机执行任务的顺序。通过这种编码方法，粒子进行初始化时即可生成一个确定的

任务分配计划和无人机的路径规划结果。

3.2 基于约束处理的可行解初始化方法

由于M-CMTAP模型包含多种约束条件，使用基于任务分配和路径规划的混合变量编码方法可能会产生不可行解。因此，本文进一步结合求解问题的特性，提出了基于约束处理的可行解初始化方法来生成可行粒子。

在初始化时，为了使每个任务分配结果、资源消耗结果和每架无人机执行任务时的路径规划结果满足约束条件，粒子的位置向量将从前往后生成。无人机和无人机执行的任务将随机地从所有任务集合和所有无人机集合中选取。如果被选取的无人机和任务满足约束条件，则向当前位置向量添加一个维度，用于表示新的任务分配结果和资源消耗结果。相应地，算法也将更新所有无人机及任务的状态。在算法后续迭代过程中，根据更新后的无人机及任务的状态来随机生成新的任务分配结果和资源消耗结果。

对应前述实例(图1所示)，粒子初始化时可选择的任务集合和无人机集合可表示为图2。

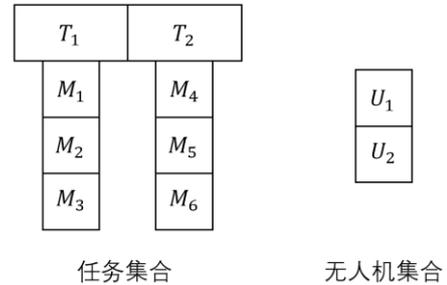


图2 任务集合和无人机集合

此时，粒子的位置向量对应的生成过程如下：首先，算法随机地选取了 M_1 并分配给侦察无人机 U_1 执行，无人机将对目标观测25.5个单位时间。然后，算法随机选取了目标 T_2 的观测任务 M_4 ，侦察无人机 U_1 被选择执行该任务， U_1 将在该任务处观测13.5个单位时间。之后，位置向量的第三列和第四列表示了两个目标上的打击任务的分配结果和资源消耗结果，战斗无人机 U_2 被选择用来依次执行这两个打击任务 M_2 和 M_5 。最后，两个目标上的打击结果评估任务 M_3 和 M_6 被分配给侦察无人机 U_1 执行， U_1 执行打击结果评估任务 M_3 和 M_6 的评估时间分别为5.5和16.5个单位时间。

使用基于约束处理的可行解初始化方法生成粒子位置向量的过程如下：

Step1: 根据任务状态和无人机状态判断任务分配结果和资源消耗结果是否满足无人机最大航程约束、战斗无人机最大携弹数约束。目标上的任务

将按照完成顺序依次分配给无人机执行以满足任务执行的时序约束。

对于上述实例，最初只有目标 T_1 的观测任务 M_1 和目标 T_2 的观测任务 M_4 可以被选择执行。

Step2: 根据任务类型随机选取对应种类的无人机，以满足无人机类型约束。

上述实例中，假设执行打击任务 M_2 和 M_5 的无人飞机为战斗无人机 U_2 ，则当前只能选择 U_2 。

Step3: 当某个任务被无人机执行完毕，对当前任务状态和无人机状态进行更新。更新任务状态时需要更新任务完成所消耗的资源以及任务完成状态。

上例中，由于 M_1 完成所需的观测时间资源小于 25.5 个单位时间，因此更新 M_1 为已完成状态，并将 M_1 从目标 T_1 的任务集合中去除。

更新无人机状态时需要更新无人机已飞行航程和无人机当前位置，由于无人机执行新任务，无人机已飞行航程将增加，无人机当前位置被更新为新任务所处位置。通过对无人机状态和任务状态的不断迭代更新，最终生成一个满足所有约束条件的可行粒子。

算法 1 展示了使用基于约束处理的可行解初始化方法生成一个粒子位置向量的过程。为了满足任务执行的时序约束，每个目标上的观测任务、打击任务和评估任务将按照顺序执行。第 2 行表示算法将依次选择目标当前需要执行的任务编号加入位置向量。第 3 行到第 9 行展示了执行该任务的无人机的选择过程。为了满足无人机类型约束，对于观测任务和评估任务，算法将随机选择一个侦察无人机执行，对于打击任务，算法则会随机选择一架战斗无人机。第 10 行表示将无人机编号和任务的资源消耗添加到位置向量中。至此，一个满足约束的无人机任务分配结果和任务资源消耗结果已生成并被保存在粒子的位置向量中。第 11 行到第 18 行展示了无人机状态和目标任务状态的更新过程，当任务完成时，该任务从目标任务集合中删除，目标任务集合为空则代表对于该目标的军事任务已全部完成。当所有目标上的任务全部完成，则代表一个可行的任务分配计划已完全生成，即一个可行粒子被成功生成。

算法 1. 基于约束处理的可行解初始化方法。

输入：目标集合 T ，任务集合 $TaskSet$ ，无人机集合 U ，位置向量 $P=(MN,UN,C)$

输出：粒子的位置向量 $P=(MN,UN,C)$

1: WHILE 目标集合 $T \neq \phi$ DO

2: 从目标集合 T 中随机选择一个目标 T_j ，从该目标的任

务集合 $TaskSet_j$ 中选择当前需要执行的任务 M_k ，将 k 添加到任务编号向量 MN 中

3: IF M_k 的类型 $MT_k \in \{Observe, Evaluate\}$ THEN

4: 从无人机集合 U 中随机选择一架侦察无人机 U_i

5: 随机生成无人机 U_i 执行任务 M_k 的时间消耗 C_i^k

6: ELSE

7: 从无人机集合 U 中随机选择一架战斗无人机 U_i

8: 随机生成无人机 U_i 执行打击任务 M_k 的导弹消耗 C_i^k

9: END IF

10: 将 t 添加到无人机编号 UN 中，将 C_i^k 添加到资源消耗 C 中

11: 更新任务 M_k 完成所需资源 $RS_k \leftarrow RS_k - C_i^k$

12: IF $RS_k \leq 0$ THEN

13: $TaskSet_j \leftarrow TaskSet_j - M_k$

14: END IF

15: IF $TaskSet_j == \phi$ THEN

16: $T \leftarrow T - T_j$

17: END IF

18: END WHILE

19: RETURN P

上述可行解初始化方法有如下优势：1) 粒子的位置向量不仅表示一个完整可行的任务分配结果，也可表示无人机执行任务时的路径规划结果；2) 该方法迭代生成新的任务分配计划前都会判断当前无人机的飞行状态和任务完成状态，因此可以确保新生成的任务分配结果和无人机资源消耗能够满足约束；3) 每次任务分配完毕都会及时更新无人机和任务的状态，因此可以迅速判断接下来生成的任务分配结果是否满足约束；4) 由于该初始化方法严格按照任务执行顺序依次分配任务，在某个任务的前置任务没有完成前，该任务不会被分配给当前无人机执行，因此生成的任务分配计划不会存在死锁状态 **Error! Reference source not found.**，即存在至少一架无人机因前置任务未执行完毕而在任务上空无限期地忙等待；5) 当模型需要考虑新的约束条件时，该方法能根据新的约束条件高效地生成可行解，具有较高的灵活性。

3.3 基于结构学习的重组方法

由于 C-MOPSO 采用了基于任务分配和路径规划的混合变量编码方式，传统的重组方法并不适合用来生成粒子，为了有效避免重组过程中生成不可行粒子，本文结合粒子的历史信息，进一步提出了基于结构学习的重组方法。

基于结构学习的重组方法生成子代粒子的过程类似于前述初始化可行解的过程，不同之处在于可行解初始化是从无到有地在粒子的位置向量中添加任务分配结果和资源消耗结果，而基于结构学习的重组方法则有效利用历史解的信息来生成新粒子。首先，通过学习父代的位置向量的结构信息，依次添加新的满足约束的任务编号、无人机编号和资源消耗值，直到所有任务已分配给无人机并被执行完成。具体过程如算法 2 所示。

算法 2. 基于结构学习的重组方法.

输入：父代个体的位置向量 $P=(MN^p, UN^p, C^p)$ ，目标集合 T ，目标的集合 $TaskSet$ ，无人机集合 U

输出：子代粒子的位置向量 $O=(MN^o, UN^o, C^o)$

- 1: 随机生成子代学习父代结构的长度 S_l ，初始化子代粒子的位置向量 $O=(MN^o, UN^o, C^o)$
- 2: FOR 位置向量维度 s , $1 \leq s \leq S_l$ DO
- 3: 任务编号 $k=MN_s^p$ ，无人机编号 $i=UN_s^p$ ，目标编号为 j
- 4: $MN_s^o \leftarrow k$, $UN_s^o \leftarrow i$, $C_s^o \leftarrow C_s^p$ //子代父代个体的部分结构
- 5: 更新第 k 个任务完成所需资源消耗 $RS_k \leftarrow RS_k - C_s^o$
- 6: IF $RS_k \leq 0$ THEN
- 7: 从第 j 个目标的任务集合 $TaskSet_j$ 中去掉任务 M_k
- 8: END IF
- 9: IF $TaskSet_j = \phi$ THEN
- 10: $T \leftarrow T - T_j$
- 11: END IF
- 12: END FOR
- 13: $s = S_l$
- 14: WHILE 目标集合 $T \neq \phi$ DO
- 15: $s \leftarrow s + 1$
- 16: 从目标集合 T 中随机选择一个目标 T_j ，从该目标的任务集合 $TaskSet_j$ 中选择当前需要执行的任务 M_k ， $UN_s^o \leftarrow i$
- 17: 从无人机集合 U 中随机选择一架侦察无人机 U_i 执行任务 M_k ， $UN_s^o \leftarrow i$
- 18: 随机生成无人机 U_i 执行任务 M_k 的资源消耗 C_i^k ， $C_s^o \leftarrow C_{ik}$
- 19: 更新任务状态、无人机状态、更新目标集合 T
- 20: END WHILE
- 21: RETURN $O=(MN^o, UN^o, C^o)$

算法 2 第 1 行到第 12 行表示子代粒子通过学习父代粒子结构中优秀搜索信息从而生成子代粒子新的结构的过程。由于保留的结构反映了一部分任务分配结果和资源消耗结果，因此，需要更新无人机状态和目标上的任务状态以方便算法在接下来生成新的任务分配结果和资源消耗结果时进行约束条件判断。第 14 行到第 20 行表示子代粒子基

于父代粒子部分结构的生成新结构的过程，通过该重组方法生成的子代保留了父代粒子部分的任务分配结果和资源消耗结果。

需要注意的是，子代粒子仅保留了父代粒子的部分结构，其保存的结构长度为 S_l 反映了子代粒子的学习效率。当子代粒子学习的父代粒子的结构长度 S_l 较大时，子代粒子相对于父代粒子有略微的变化，子代粒子将在父代粒子基础上进行局部搜索，而当 S_l 的数值较小时，子代粒子与父代相比变化较大，子代种群的多样性将得到提升。为了平衡多样性和收敛性，本文的结构长度 S_l 随父代粒子位置向量长度动态变化，设置为父代粒子位置向量长度的一半。

上述重组方法有效地利用了父代粒子的结构信息，重组后生成的新粒子同样可表示唯一确定的无人机任务分配结果和执行任务时的路径规划结果。同时该方法在生成一组任务分配结果和资源消耗结果前会对无人机、任务的状态进行判断，因此可以保证最终生成的任务分配计划能够满足约束条件。

3.4 基于协同进化的种群更新策略

为了进一步加速算法收敛，提升算法性能，本文提出一种基于协同进化机制的种群更新策略，从种群中选择较优个体来更新历史最优种群 P_{best} 。

在大部分的多目标优化算法中，选择优秀个体主要有三种方法：基于帕累托支配 (Pareto Dominance) 的方法 Error! Reference source not found.、基于分解的方法 Error! Reference source not found. 和基于指标的方法 Error! Reference source not found. [19] [20]。基于帕累托支配的方法通过比较各个目标函数上的大小关系确定解的支配关系，算法从种群中选择非支配的个体作为优秀个体。基于分解的方法能够将解的所有目标值标量化，通过比较标量化后的数值大小，算法可以方便地从种群中选择出优秀个体，然而采用该方法的算法在求解帕累托前沿 (Pareto Front, PF) 不均匀或较复杂的多目标优化问题时，种群的多样性可能会变差 Error! Reference source not found.。基于指标的方法通过特定的指标计算种群在目标空间的状态，这些指标通常反映了种群的多样性和收敛性，如 HyperVolume 指标 Error! Reference source not found. 和 IGD 指标 Error! Reference source not found.，通过比较指标大小，算法可以选择出多样性和收敛性较好的个体，但该类算法需要消耗额外的计算资源。例如，在计算 HyperVolume 指标时，消耗的计算资源会随着优化目标数目增加而快速增长。

基于上述分析，本文采用基于帕累托支配的方法选择最优个体。对于大小为 $2N$ 的种群，首先采用快速非支配排序方法以及拥挤度距离指标 Error!

Reference source not found. (Crowding Distance) 从中选择出 N 个帕累托占优的个体, 并基于当前 P_{best} 更新粒子群 Pop 。考虑到算法迭代过程中产生的后 N 个非占优的个体同样保存了一定的历史搜索信息, 为了充分利用这些信息加快收敛, 引入协同进化机制, 基于后 N 个非占优个体重组生成一个与 P_{best} 协同进化的种群 $Co-Pop$ 。通过将 $Co-Pop$ 与 Pop 进行融合, 构成新的大小为 $2N$ 的种群。通过采用基于帕累托支配的方法选择占优个体直至算法满足停止条件。具体更新过程如算法 3 所示。

在每一轮迭代时首先根据上一轮中的前 N 个占优个体采用基于结构学习的重组方法生成粒子群 Pop , 同时利用后 N 个非占优个体采用结构学习的重组方法生成一个协同种群 $Co-Pop$ 。通过这两个种群 $Co-Pop$ 与 Pop 竞争来更新粒子群, 采用快速非支配排序方法, 从 Pop 和 $Co-Pop$ 组成的融合种群 $Combine_Pop$ 中选取前 N 个占优个体更新 P_{best} 。协同种群 $Co-Pop$ 将与粒子群 Pop 共同进化直到法迭代结束。在上述迭代过程中, $Co-Pop$ 为 Pop 提供了竞争压力, $Co-Pop$ 中的优秀个体将取代原有的 Pop 中的个体 (通过快速非支配排序), 从而获得历史最优种群 P_{best} 。同时, $Co-Pop$ 与 Pop 协同进化时也为 P_{best} 的更新提供了额外的搜索信息, 提高了算法搜索效率。

算法 3. 基于协同进化的种群更新策略。

输入: 协同种群 $Co-Pop$, 粒子群 Pop 以及历史最优种群 P_{best}

输出: 最优种群 P_{best}

```

1: WHILE 不满足停止条件 DO
2:   选取  $N$  个占优个体, 基于当前  $P_{best}$  采用结构重组方法生成粒子群  $Pop$ 
3:   选取  $N$  个非占优个体, 采用结构重组方法生成协同进化种群  $Co-pop$ 
4:   生成融合种群  $Combine\_Pop \leftarrow Co-Pop \cup Pop$ 
5:    $quickNondominatedSort(Combine\_Pop, 2N)$ 
6:   更新  $P_{best}$ ,  $Co-Pop$ 
7: END WHILE
8: RETURN  $P_{best}$ 

```

与传统的种群更新策略不同, 上述种群更新策略不仅利用了占优个体的信息, 还充分利用了非占优个体的信息, 从而有效增加了种群多样性。同时, 由于 P_{best} 保存帕累托占优个体, 基于 P_{best} 生成的 Pop 与 $Co-Pop$ 融合生成的新种群也能较好地保留种群的收敛性信息。因此, 上述策略可以实现种群多样性和收敛性的有效平衡, 有助于生成更优秀的

粒子, 从而获得更好的解。

4 仿真实验

为了验证 C-MOPSO 求解 M-CMTAP 的有效性, 本文基于不同的任务类型和无人机分布情况设计了 4 个 M-CMTAP 模型实例并进行仿真实验。

4.1 实例设计

基于 M-CMTAP 模型, 本文设计了 4 个模型的实例。每一个实例包含 14 个无人机和 6 个军事任务目标, 它们被随机地设置在一个固定位置。每个无人机的属性包括飞行速度、最大携弹数目和最远飞行距离。每个目标的三种任务 (观测任务、打击任务和打击结果评估任务) 的属性分别为任务完成所需要的观测时间、打击导弹数目或评估时间。

基于不同的任务和无人机分布情况生成实例的属性设置各不相同。实例 1 中各个无人机、目标和任务的属性设置情况如表 4 和表 5 所示, 实例 2、实例 3 和实例 4 的属性的详细内容展示在附录 A 中。相比于实例 3 和实例 4, 实例 1 和实例 2 中的目标和无人机在军事区域内分布地相对集中。这意味着路径规划时无人机可选择飞行的次优航线较多, 在任务分配时可选择的合适的无人机较多。因此, 在实例 1 和实例 2 的目标空间中存在有较多的局部最优解。算法在实例 1 和实例 2 上的实验结果能反映算法求解多峰优化问题时的性能, 算法在实例 3 和实例 4 上的实验结果能反映算法求解 M-CMTAP 模型的收敛性能。通过在四个实例上做实验, 可以评估算法的探索能力和勘探能力。

在 M-CMTAP 模型中, 针对每个目标的军事任务包含观测任务、打击任务和评估任务。无人机执行任务时需要赶到任务对应的目标所在的坐标处, 坐标单位为千米。完成观测任务和评估任务需要侦察无人机在目标处执行一段时间 (时间单位为秒)。完成打击任务则需要消耗战斗无人机的机载导弹。

在每个实例中, 异构的无人机系统包括 8 架侦察无人机和 6 架战斗无人机。战斗无人机为了执行打击任务需要携带一定的导弹, 而侦察无人机则不携带导弹。所有无人机的飞行速度如表所示。此外, 每架无人机的最远飞行距离设为 1000 千米。

表 4 实例 1 目标属性数值

属性 目标	坐标	任务	任务完成条件
目标 1	(13,41)	观测任务 M_1	100s
		打击任务 M_2	2
		评估任务 M_3	40s
目标 2	(61,83)	观测任务 M_4	120s
		打击任务 M_5	1
		评估任务 M_6	20s
目标 3	(79,12)	观测任务 M_7	40s
		打击任务 M_8	3
		评估任务 M_9	10s
目标 4	(41,98)	观测任务 M_{10}	90s
		打击任务 M_{11}	4
		评估任务 M_{12}	15s
目标 5	(23,65)	观测任务 M_{12}	150s
		打击任务 M_{14}	2
		评估任务 M_{15}	20s
目标 6	(53,19)	观测任务 M_{16}	20s
		打击任务 M_{17}	2
		评估任务 M_{18}	5s

表 5 实例 1 无人机属性数值

属性 无人机	坐标	无人机类型	速度 (km/s)	携弹数目
无人机 1	(13,86)	侦察无人机	0.1	0
无人机 2	(53,47)	侦察无人机	0.12	0
无人机 3	(23,91)	侦察无人机	0.12	0
无人机 4	(65,31)	侦察无人机	0.11	0
无人机 5	(64,24)	侦察无人机	0.1	0
无人机 6	(74,12)	侦察无人机	0.12	0
无人机 7	(35,12)	侦察无人机	0.11	0
无人机 8	(39,82)	侦察无人机	0.11	0
无人机 9	(47,46)	战斗无人机	0.16	5
无人机 10	(15,78)	战斗无人机	0.10	10
无人机 11	(63,83)	战斗无人机	0.12	6
无人机 12	(23,42)	战斗无人机	0.13	7
无人机 13	(96,14)	战斗无人机	0.09	5
无人机 14	(47,25)	战斗无人机	0.11	7

4.2 对比算法和评价指标

为了验证 C-MOPSO 的性能，本文选取了目前具有代表性的 3 个基于协同进化策略的算法进行对比实验，包括：CPSO^{Error! Reference source not found.}、CoMOLS/D^{Error! Reference source not found.}和 CMPSO^{Error! Reference source not found.}。其中，CMPSO 和 CPSO 在迭代时保存了多个粒子群，单个粒子群负责优化单个目标并与其他粒子群协同进化最终求解整个多目标优化问题。相比于 CMPSO，CPSO 采用了瓶颈学

习策略，该策略能够帮助粒子群在优化单个目标的同时关注其他目标的情况，提高了整个种群的多样性。CoMOLS/D 则基于分解方法更新种群，第一种群 P 采用权重和法 (Weighted Sum, WS) 选择个体，而与种群 P 协同进化的第二种群 Q 基于反转的边界交叉惩罚^{Error! Reference source not found.}(Inverted Penalty-based Boundary Intersection, iPBI)选择个体。第二种群 Q 采用的权重向量是根据第一种群 P 的分布协同进化生成的。因此，种群 Q 弥补了 P 在目标空间分布上存在的间隙，使得整个种群分布地更加均匀。

为了保证实验的公平性，种群大小 N 均设置为 500。此外，CPSO 和 CMPSO 采用一个外部集合保存多个粒子群中的非支配解，外部集合的大小 NA 被设置为 250。CoMOLS/D 中的第一种群 P 和第二种群 Q 的个体总数 N 为 500，第二种群 Q 的大小 L 固定为 250。第一种群 P 中每个个体的邻居数目 T 为 5，第二种群 Q 采用基于 iPBI 的分解方法选择个体，iPBI 中的惩罚系数 θ 设置为 5.0。各算法的参数设置如表 6 所示。

表 6 算法参数设置

算法名称	参数设置
C-MOPSO	$N=500$
CPSO	$N=500$ $NA=250$
CMPSO	$N=500$ $NA=250$
CoMOLS/D	$N=500$ $L=250$ $\theta=5.0$ $T=5$

需要指出的是，为了使选取的各对比算法能够求解本文的 M-CMTAP 模型，上述对比算法中采用了与 C-MOPSO 相同的初始化和重组方法。因此，将本文提出的 C-MOPSO 与上述对比算法对比，不仅能够验证本文提出的基于协同进化的种群更新策略的性能，还能进一步验证本文提出的初始化方法和重组方法的有效性。

本文选取常用的多目标解集评价指标 HyperVolume (HV)^{Error! Reference source not found.}来评估算法的性能优劣。HV 指标描述了算法求得的非支配解集与参考点构成的超立方体的体积大小。当非支配解集越靠近真实的 PF、分布地越均匀，HV 值就越大，反之则越小。

4.3 实验结果与分析

为了保证仿真实验结果的准确性，每个算法将在每个实例上单独运行 30 次，算法的平均结果 (MEAN) 为 30 次运行结果的平均值，算法求得的最优结果 (BEST) 和最差结果 (WORST) 为 30 次运行结果中的最优值和最差值。每次运行时，算法的最大迭代次数设置为 300。

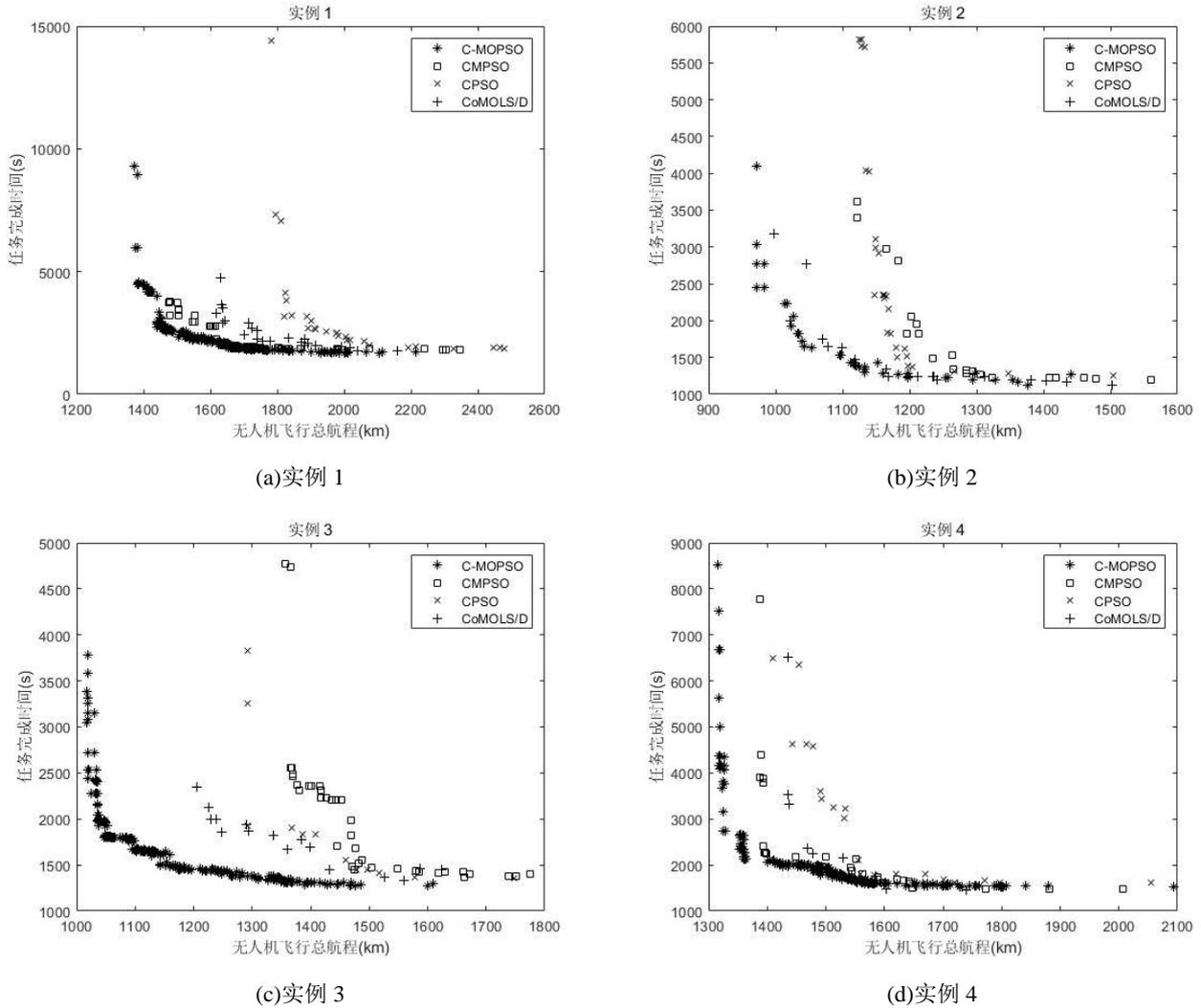


图3 算法在测试实例上的解集分布图对比

C-MOPSO 与对比算法在每个实例上所求得解集的分布情况如图 3 所示。星号点表示 C-MOPSO 求得的解集，正方形点表示 CMPSO 求得的解集，叉号点表示 CPSO 求得的解集，加号点表示 CoMOLS/D 求得的解集。从图 3 可以明显看出，C-MOPSO 在所有测试实例上求得的解集更加靠近问题真实的 PF，其他三个对比算法 CPSO，CMPSO 和 CoMOLS/D 则表现地劣于 C-MOPSO。在实例 1 上，C-MOPSO 求得的解集分布均匀且更加逼近真实 PF，且 CMPSO 的收敛性较好，但 CMPSO 求得的解集在分布图的中间部分出现缺失，解集的多样性一般，原因可能是 CMPSO 中的每个粒子群只优化一个目标，多个目标中间的区域很少得到所有粒子群的关注。CPSO 和 CoMOLS/D 求得的解集分布较为均匀但收敛性劣于 C-MOPSO。在实例 2 上，

C-MOPSO 和 CoMOLS/D 表现出了相似的收敛性，两个算法求出的解集分布地相对均匀。CPSO 和 CMPSO 求得的解集分布地较为均匀，但解集的收敛性较差。在实例 3 和实例 4 上，C-MOPSO 表现出了明显的优势，其求得的解集不仅保持了较好的多样性，而且更加逼近真实 PF，而其他三个对比算法表现较差，尽管 CMPSO 和 CoMOLS/D 求得的解集的收敛性不错，但解集分布的均匀性不够好。

表 7 为各算法在不同测试实例上的实验结果。MEAN 表示算法运行 30 次得到的 HV 的平均值，反映了算法求得的非支配解集的收敛性和多样性。BEST 表示算法运行 30 次后得到的最优的 HV 值，WORST 表示算法得到的最差 HV 值，反映了算法性能的稳定性。所有算法求得的最优结果用黑色背景标注，次优结果用灰色背景标注。

表 7 算法 HV 结果对比

		CPSO ^{Error!} Reference source not found.	CMPSO ^{Error!} Reference source not found.	CoMOLS/D ^{Error!} Reference source not found.	C-MOPSO
实例 1	MEAN	2.24E+05	1.69E+05	1.66E+05	5.55E+05
	BEST	3.75E+05	2.96E+05	2.86E+05	8.92E+05
	WORST	1.03E+05	6.01E+04	7.03E+04	1.11E+05
实例 2	MEAN	3.73E+05	1.72E+05	1.20E+05	5.19E+05
	BEST	6.65E+05	2.56E+05	2.21E+05	8.85E+05
	WORST	1.96E+05	1.14E+05	8.04E+04	2.21E+05
实例 3	MEAN	2.36E+05	1.79E+05	9.47E+04	5.13E+05
	BEST	4.04E+05	4.67E+05	1.89E+05	9.02E+05
	WORST	1.46E+05	8.04E+04	6.02E+04	1.42E+05
实例 4	MEAN	3.30E+05	1.75E+05	1.16E+05	5.50E+05
	BEST	6.44E+05	2.36E+05	1.89E+05	9.13E+05
	WORST	1.50E+05	8.30E+04	7.62E+04	2.44E+05

从表 7 可以明显看出，C-MOPSO 在所有实例上求得解集的 HV 值均大于其他对比算法，说明 C-MOPSO 求多次运行得的解集具有较好的多样性和收敛性。CPSO 在所有实例上取得了第二的结果，CoMOLS/D 和 CMPSO 在所有实例上的结果较差。另外，C-MOPSO 在所有实例上求得的最优结果均比其他对比算法求得的最优结果好，且 C-MOPSO 在所有实例上求得的最差结果均比其他对比算法求得的最差结果好，这也进一步说明 C-MOPSO 性能较为稳定。CPSO 比 CMPSO 表现较好的原因可能是 CPSO 采用的瓶颈学习策略保证每个粒子群在优化单个目标的同时关注其他优化目标，因此 CPSO 解决了 CMPSO 粒子分布不均匀的问题，求得的解集较好地保持了多样性。而 CoMOLS/D 采用权重和法选择第一种群 P 中的优秀解，但由于权重和法并不适合求解 PF 形状为下凸的问题，因而 CoMOLS/D 求得解集的多样性并不好。总之，C-MOPSO 能够有效地求解上述 M-CMTAP 实例，相比于其他算法，C-MOPSO 求得的解集较好地保持了多样性和收敛性。

C-MOPSO 表现更优主要归因于以下四个方面：首先，C-MOPSO 采用的基于任务分配和路径规划的编码方式可以帮助算法有效地解耦 M-CMTAP 中两个重要的优化阶段，即任务分配阶段和路径规划阶段。基于此编码方法生成的粒子可以表示确定的任务分配结果和无人机路径规划结果。其次，基于该编码方式的可行解初始化方法可

以有效地处理约束并生成满足约束条件的可行解。因此 C-MOPSO 在迭代开始时就能保存充足的可行粒子，这些粒子保存的可行解的结构信息也是种群接下来的进化基础。此外，C-MOPSO 维护的协同种群 Co-Pop 充分地利用了快速非支配排序过程中丢弃的搜索信息，基于此信息生成、更新的 Co-Pop 增加了与粒子群 Pop 融合构成的新种群的多样性。另外，基于结构学习的重组方法通过粒子群的历史最优种群 Pbest 重组生成新的粒子群，使得父代中优秀粒子的部分结构得以保留，同时子代粒子基于这些结构添加了新的任务分配和路径规划结果，有效保证了更新后粒子群的收敛性和多样性。

综上所述，本文所提出的 C-MOPSO 算法能够有效求解 M-CMTAP 问题，同时，与其他采用协同进化策略算法的对比实验结果也进一步验证了本文提出的协同进化策略有助于提升算法求解的收敛性和多样性。

5 总结与展望

为了更加真实客观地描述无人机协同作战场景，本文建立了一个考虑多种约束条件和多个优化目标的无人机协同多任务分配模型，M-CMTAP。该模型充分考虑了军事场景中的实际需求，以最少化无人机总航程和最小化任务完成时间为两个优化指标，并同时考虑了无人机资源约束、无人机类型约束、任务执行时序约束和多机协同约束等约束

条件。由于 M-CMTAP 中一个完整任务分配计划包含了任务分配结果和无人机执行任务时的资源消耗结果,因此, M-CMTAP 的决策变量既包含代表任务下标的离散变量,又包含了表示资源消耗的连续变量以及离散变量。同时, M-CMTAP 中的多种约束既有不等式约束,也有等式约束。这些特性使得问题对应的解空间变得不规则,增加了模型求解难度。传统的多目标优化算法并不适合求解该 M-CMTAP 模型。

为了高效地求解该 M-CMTAP 模型,本文引入协同进化思想提出了 C-MOPSO。首先,针对模型中的混合变量,本文提出了基于任务分配和路径规划的混合编码方法。一个粒子的位置向量包含任务下标、无人机下标和资源消耗三个部分。三个部分的每个维度一一对应,表示无人机执行相应任务时的资源消耗,即一个任务分配结果和资源消耗结果。同时,任务在粒子位置向量中出现的顺序被定义为无人机的执行顺序。因此当一个完整的粒子被生成时,一个可行的任务分配结果和路径规划结果也就生成了。在此基础上,本文提出了基于约束处理的可行解初始化方法,该方法按照任务的执行顺序将需要被执行的任务随机地分配给符合要求的无人机执行,在一个任务分配结果和资源消耗结果被添加到位置向量前,会首先判断无人机和任务的状态是否满足约束条件。因此,该方法能够提前判断约束以生成可行解。此外,为了生成更加优秀的粒子,本文提出的基于结构学习的重组方法通过学习优秀父代粒子的结构,从而保证生成的子代粒子可以保留优秀父代粒子的部分结构,进而保证种群的收敛性和多样性。另外,提出了一种基于协同进化的种群更新策略,该策略通过协同种群和原有种群的竞争协同来更新粒子群,实现了种群多样性和收敛性之间的有效平衡。

在多个实例上的对比实验结果表明了 C-MOPSO 求解得到的非支配解集在保持较好收敛性的同时还能维持较好的多样性。因此,可以得出结论,本文提出的 C-MOPSO 算法能够有效求解 M-CMTAP 问题。此外,得益于 C-MOPSO 灵活地编码方式和初始化方法, C-MOPSO 可以被扩展应用到更多类似问题的求解中。

未来,将考虑更多的实际情况来进一步改进现有模型,如针对任务过多导致无人机由于资源受限无法完成所有任务的情形构建模型。同时,随着人工智能技术的发展,一些新的机器学习方法如深度

学习、深度强化学习,其理论成果也为智能优化算法的研究提供了新的思路。下一步研究工作将结合这些新的技术理论方法来改进现有算法,以进一步提高无人机协同多任务分配问题的求解性能。

参考文献

- [1] Li De-Ren, Li Ming. Research progress and application prospects of UAV remote sensing system, *Geomatics and Information Science of Wuhan University*, 2014, 39(05):505-513. (in Chinese)
(李德仁,李明.无人机遥感系统的研究进展与应用前景.武汉大学学报(信息科学版),2014,39(05):505-513.)
- [2] Edison E, Shima T. Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers & Operations Research*, 2011, 38(1):340-356.
- [3] Huang L, Qu H, Zuo L. Multi-type UAVs cooperative task allocation under resource constraints. *IEEE Access*, 2018, 6: 17841-17850.
- [4] Cheng Q, Yin D, Yang J, et al. An auction-based multiple constraints task allocation algorithm for multi-UAV system// *Proceedings of the IEEE 2016 International Conference on Cybernetics, Robotics and Control (CRC)*. Hong Kong, China, 2016: 1-5.
- [5] Ye, F., Chen, J., Tian, Y., & Jiang, T. Cooperative Task Assignment of a Heterogeneous Multi-UAV System Using an Adaptive Genetic Algorithm. *Electronics*, 2020, 9(4), 687.
- [6] Guangtong X, Li L, Long T, et al. Cooperative Multiple Task Assignment Considering Precedence Constraints Using Multi-Chromosome Encoded Genetic Algorithm// *Proceedings of the 2018 AIAA Guidance, Navigation, and Control Conference*. Kissimmee, Florida, USA, 2018, 1859.
- [7] Su Fei, Chen Yan, Shen Lin-Cheng. UAV Cooperative Multi-task Assignment Based on Ant Colony Algorithm. *Acta Aeronautica ET Astronautica Sinica*, 2008, (S1):184-191. (in Chinese)
(苏菲,陈岩,沈林成.基于蚁群算法的无人机协同多任务分配[J].航空学报,2008(S1):184-191.)
- [8] Liang Guo-Qiang, Kang Yu-Hang, Xing Zhi-Chuan, Yin Gao-Yang. UAV Cooperative Multi-task Assignment Based on Discrete Particle Swarm Optimization Algorithm. *Computer Simulation*, 2018, 35(02):22-28. (in Chinese)
(梁国强,康宇航,邢志川,尹高扬.基于离散粒子群优化的无人机协同多任务分配[J].计算机仿真,2018,35(02):22-28.)
- [9] Wang Fu-Xing. Multiple unmanned aerial vehicles self-organizing control based on particle swarm optimization[M.S. Thesis]. Nanjing University of Posts and Telecommunications, Nanjing, 2019. (in Chinese)
(王福星.基于粒子群优化算法的多无人机自组织控制[硕士学位论文].南京邮电大学,南京,2019.)
- [10] L. Cao, H. shun Tan, H. Peng and M. cong Pan. Multiple UAVs hierarchical dynamic task allocation based on PSO-FSA and

decentralized auction// Proceedings of 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). Bali, Indonesia, 2014: 2368-2373.

[11] WDi Bin, Zhou Rui, Ding Quan-Xin. Distributed coordinated heterogeneous task allocation for unmanned aerial vehicles. Control and Decision, 2013,28(02):274-278. (in Chinese)
(邸斌,周锐,丁全心.多无人机分布式协同异构任务分配.控制与决策,2013,28(02):274-278.)

[12] Minimum-risk problem of unmanned aerial vehicle task allocation with expert belief degree. Control and Decision, 2019,34(09):2036-2040. (in Chinese)
(王健,郭建胜,慕容政,毛声,顾涛勇.带有专家信度的无人机任务分配最小风险问题.控制与决策,2019,34(09):2036-2040)

[13] Han Bo-wen, Yao Pei-yang, Sun Yu. UAVS Cooperative Task Allocation Based on Multi-Objective MSQPSO Algorithm. Acta Electronica Sinica, 2017, 45(8): 1856-1863. (in Chinese)
(韩博文,姚佩阳,孙昱.基于多目标MSQPSO算法的UAVS协同任务分配.电子学报,2017,45(8):1856-1863.)

[14] Tian Zhen, Wang Xiao-Fang. UAVS Cooperative multiple task assignment for heterogeneous multi-UAVs with multi-chromosome genetic algorithm. Flight Dynamics,2019, 37(01):39-44. (in Chinese)
(田震,王晓芳.基于多基因遗传算法的异构多无人机协同任务分配.飞行力学,2019,37(01):39-44.)

[15] Wang Ling, Wang Sheng-Yao, Deng Jin. Advances in co-evolutionary algorithms. Control and Decision, 2015,30(02):193-202. (in Chinese)
(王凌,沈婧楠,王圣尧,邓瑾.协同进化算法研究进展.控制与决策,2015,30(02):193-202.)

[16] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[17] Zhang Qingfu, Li Hui. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation, 2008, 11(6):712-731.

[18] Beume N , Naujoks B , Emmerich M . SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research, 2007, 181(3):1653-1669.

[19] Sun Y , Yen G G , Yi Z . IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems. IEEE Transactions on Evolutionary Computation, 2019, 23(2):173-187.

[20] Coello C A C, Sierra M R. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm//Proceedings of Mexican International Conference on Artificial Intelligence. Mexico City, Mexico, 2004: 688-697.

[21] Li Hui, Zhang Qingfu. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation, 2009, 13(2):284-302.

[22] Liu X F , Zhan Z H , Gao Y , et al. Coevolutionary Particle Swarm Optimization With Bottleneck Objective Learning Strategy for Many-Objective Optimization. IEEE Transactions on Evolutionary

Computation, 2018, 23(4) : 587-602.

[23] Cai X , Hu M , Gong D , et al. A decomposition-based coevolutionary multiobjective local search for combinatorial multiobjective optimization. Swarm & Evolutionary Computation, 2019, 49: 178-193.

[24] Zhan Z H , Li J , Cao J , et al. Multiple Populations for Multiple Objectives: A Coevolutionary Technique for Solving Multiobjective Optimization Problems. IEEE Transactions on Cybernetics, 2013, 43(2):445-463.

附录A.

实例 2 目标属性数值

目标	属性		
	坐标	任务	任务完成条件
目标 1	(74,29)	观测任务 M_1	100s
		打击任务 M_2	2
		评估任务 M_3	40s
目标 2	(68,33)	观测任务 M_4	120s
		打击任务 M_5	1
		评估任务 M_6	20s
目标 3	(76,38)	观测任务 M_7	40s
		打击任务 M_8	3
		评估任务 M_9	10s
目标 4	(96,26)	观测任务 M_{10}	90s
		打击任务 M_{11}	4
		评估任务 M_{12}	15s
目标 5	(52,55)	观测任务 M_{12}	150s
		打击任务 M_{14}	2
		评估任务 M_{15}	20s
目标 6	(59,98)	观测任务 M_{16}	20s
		打击任务 M_{17}	2
		评估任务 M_{18}	5s

实例 2 无人机属性数值

无人机	属性			
	坐标	无人机类型	速度 (km/s)	携弹数目
无人机 1	(57,5)	侦察无人机	0.1	0
无人机 2	(42,8)	侦察无人机	0.12	0
无人机 3	(50,99)	侦察无人机	0.12	0
无人机 4	(62,25)	侦察无人机	0.11	0
无人机 5	(20,61)	侦察无人机	0.1	0
无人机 6	(83,46)	侦察无人机	0.12	0
无人机 7	(56,90)	侦察无人机	0.11	0

无人机 8	(39,45)	侦察无人机	0.11	0
无人机 9	(89,62)	战斗无人机	0.16	5
无人机 10	(95,9)	战斗无人机	0.10	10
无人机 11	(56,80)	战斗无人机	0.12	6
无人机 12	(86,46)	战斗无人机	0.13	7
无人机 13	(89,34)	战斗无人机	0.09	5
无人机 14	(53,11)	战斗无人机	0.11	7

实例 3 目标属性数值

属性 目标	坐标	任务	任务完成条件
目标 1	(36,49)	观测任务 M_1	100s
		打击任务 M_2	2
		评估任务 M_3	40s
目标 2	(70,47)	观测任务 M_4	120s
		打击任务 M_5	1
		评估任务 M_6	20s
目标 3	(25,15)	观测任务 M_7	40s
		打击任务 M_8	3
		评估任务 M_9	10s
目标 4	(62,89)	观测任务 M_{10}	90s
		打击任务 M_{11}	4
		评估任务 M_{12}	15s
目标 5	(54,42)	观测任务 M_{12}	150s
		打击任务 M_{14}	2
		评估任务 M_{15}	20s
目标 6	(61,39)	观测任务 M_{16}	20s
		打击任务 M_{17}	2
		评估任务 M_{18}	5s

实例 3 无人机属性数值

属性 无人机	坐标	无人机类型	速度 (km/s)	携弹数目
无人机 1	(73,91)	侦察无人机	0.1	0
无人机 2	(65,64)	侦察无人机	0.12	0
无人机 3	(56,37)	侦察无人机	0.12	0
无人机 4	(36,96)	侦察无人机	0.11	0
无人机 5	(20,0)	侦察无人机	0.1	0
无人机 6	(1,68)	侦察无人机	0.12	0
无人机 7	(51,74)	侦察无人机	0.11	0
无人机 8	(58,69)	侦察无人机	0.11	0
无人机 9	(73,76)	战斗无人机	0.16	5
无人机 10	(69,20)	战斗无人机	0.10	10
无人机 11	(52,46)	战斗无人机	0.12	6
无人机 12	(46,98)	战斗无人机	0.13	7

无人机 13	(92,86)	战斗无人机	0.09	5
无人机 14	(47,25)	战斗无人机	0.11	7

实例 4 目标属性数值

属性 目标	坐标	任务	任务完成条件
目标 1	(30,43)	观测任务 M_1	100s
		打击任务 M_2	2
		评估任务 M_3	40s
目标 2	(27,82)	观测任务 M_4	120s
		打击任务 M_5	1
		评估任务 M_6	20s
目标 3	(59,13)	观测任务 M_7	40s
		打击任务 M_8	3
		评估任务 M_9	10s
目标 4	(42,28)	观测任务 M_{10}	90s
		打击任务 M_{11}	4
		评估任务 M_{12}	15s
目标 5	(55,39)	观测任务 M_{12}	150s
		打击任务 M_{14}	2
		评估任务 M_{15}	20s
目标 6	(8,46)	观测任务 M_{16}	20s
		打击任务 M_{17}	2
		评估任务 M_{18}	5s

实例 4 无人机属性数值

属性 无人机	坐标	无人机类型	速度 (km/s)	携弹数目
无人机 1	(34,71)	侦察无人机	0.1	0
无人机 2	(94,92)	侦察无人机	0.12	0
无人机 3	(79,2)	侦察无人机	0.12	0
无人机 4	(5,57)	侦察无人机	0.11	0
无人机 5	(19,79)	侦察无人机	0.1	0
无人机 6	(38,61)	侦察无人机	0.12	0
无人机 7	(73,17)	侦察无人机	0.11	0
无人机 8	(11,90)	侦察无人机	0.11	0
无人机 9	(78,4)	战斗无人机	0.16	5
无人机 10	(73,71)	战斗无人机	0.10	10
无人机 11	(77,19)	战斗无人机	0.12	6
无人机 12	(86,21)	战斗无人机	0.13	7
无人机 13	(38,25)	战斗无人机	0.09	5
无人机 14	(79,4)	战斗无人机	0.11	7



Wang Feng, Ph.D., associate professor. Her research interests include evolutionary computation, machine learning and intelligent information retrieval.

Zhang Heng, graduate student. His research interests include

evolutionary computation and machine learning.

Han Meng-Chen, graduate student. His research interest is evolutionary computation.

Xing Li-Ning, Ph.D., professor. His research interest covers intelligent optimization theory, method and application.

Background

The UAV system composed by multiple UAVs has been widely applied in military strikes, scientific research and observation. The UAV cooperative multi-task allocation problem model (CMTAP) describes the scenario of assigning tasks to UAVs and UAV path planning. Studying the algorithm to solve this model can help researchers to dispatch UAVs more scientifically and reasonably. However, solving this model is challenging. On the one hand, the model is a combinatorial optimization problem. It is difficult for existing algorithms to solve the problem effectively in a limited time; on the other hand, the model includes mixed variables, which increases the complexity of the problem space and increases the difficulty to solve this problem.

Therefore, in this paper, we introduced the co-evolution strategy and proposed a co-evolution based mixed variable multi-objective particle swarm optimization algorithm C-MOPSO. C-MOPSO adopts an encoding scheme based on task assignment and path planning to represent the task

assignment results and path planning results of the UAV and a constraint handling based feasible solution initialization method to generate feasible particles efficiently. In C-MOPSO, a structure learning based reproduction method is employed to update the particles and the co-evolutionary population retained by C-MOPSO will co-evolve with the particle swarm to update the historical optimal population. Compared with three state-of-the-art co-evolutionary based algorithms on four test cases, the experimental results show that C-MOPSO outperforms the others.

This work is supported by the National Natural Science Foundation of China (Nos. 61773296 and 61773120), the Foundation for the Author of National Excellent Doctoral Dissertation of China (No. 2014-92), and the innovation Team of Guangdong Provincial Department of Education (2018KCXTD031).